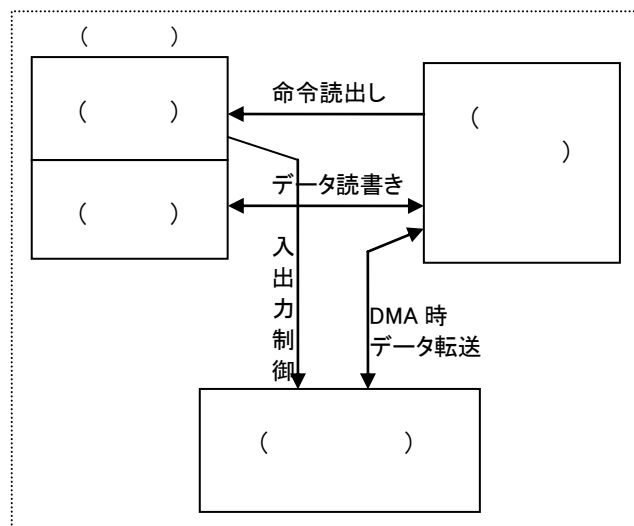


第4回 命令とその実行の仕組み 予習資料

第4回授業の前にあらかじめ目を通し、問題を解いておくこと。

[1] コンピュータの構成要素 (教科書 2.12)

- a. コンピュータの構成要素は、
 データに対して、算術演算(数の $+$ \times \div などの計算)や論理演算(AND/OR/NOT)を行う ()装置、
 データとプログラムを記憶する ()装置、
 データやプログラムを読み込んだり、処理結果を書きだしたり(表示したり)する ()装置、
 全体をコントロールする ()装置、
 から成る
- b. 演算装置と制御装置を併せて、日本語では()、英語の3文字略称で()と呼び、1つの装置としてIC化されていることが多い。
- c. b. を前提とすると、コンピュータの構成要素は、()、()、()である。
- d. CPU(中央処理装置)の構成要素は、()と()である。
- e. 右図はコンピュータハードウェアの大まかな構造を示したもののだが、それぞれの箱に名称をつけよ



- f. 次の中で正しいものはどれか
- ① CPUは中央処理装置とも呼ばれ、内部に計算をするための演算装置と、命令の実行を制御する制御装置が含まれる
 - ② CPUは中央処理装置とも呼ばれ、内部に計算をするための演算装置と、データを記憶するメインメモリが含まれる
 - ③ メモリは主記憶とも呼ばれ、演算に必要なデータ(プログラム内の変数と定数)のみが置かれる
 - ④ メモリは主記憶とも呼ばれ、データの他にプログラム(命令列)が置かれ、その命令列がメモリ内の機構で実行される
- g. 正しいものはどれか
- CPUには
- ①記憶装置と演算装置が含まれる
 - ②演算装置と制御装置が含まれる
 - ③主記憶装置と演算装置が含まれる
 - ④主記憶装置と制御装置が含まれる
- h. 演算装置は
- ①コンピュータの処理をすべて行う
 - ②コンピュータの処理に必要な数値の計算を行う
 - ③プログラムの順番を計算して決める
 - ④プログラムを、中身を解釈して実行する
- i. 制御装置は
- ①コンピュータの処理をすべて行う
 - ②コンピュータの処理に必要な数値の計算を行う
 - ③プログラムの順番を計算して決める
 - ④プログラムを、中身を解釈して実行する
- j. 演算装置は、教科書図 2.6 に示すように、()と()と()から成る。
- k. ALU とは、((英語))の略(頭文字)で、日本語では()と呼ばれ

る

l. 汎用レジスタ((英語))は、高速に動作する「置数器」である。

「置数器」とは、数を置く装置という意味で、計算した結果を一時的に保管する。算術論理演算回路は数を覚えておく機能は無いので、汎用レジスタもしくは主記憶(メインメモリ)から読み出した数値を入力として演算(たとえば足し算)をし、結果を汎用レジスタもしくは主記憶(メインメモリ)に書き込む。

「一時的に保管する」の例: A+B+Cを計算する時、A+Bの結果を汎用レジスタの1つに保管しておき、その保管しておいた値とCを加えることによって、A+B+Cを作る。

m. 汎用レジスタの個数や記憶できるデータ長(ビット数)は、(CPU) ← 教科書 2.1.4

たとえば、情報処理技術者試験での COMET II では、データ長()の汎用レジスタを ()持っている。

n. フラグレジスタは、算術論理演算回路で計算した結果について、(追加の)情報を覚えている。たとえば COMET II では ()()()の3つのビットを(真か偽かを0か1かで)覚えている。

具体的な意味は(教科書の図 2.7 を参照せよ)、OFは()、SFは()、ZFは()である。

OF: 足し算などで計算結果が15ビット(16ビットだがMSBが符号のため除く)を越えてしまった時に1が立つ。

たとえば 111 1111 1111 1111 + 000 0000 0000 0001 は、15ビットでは収まらなくなる。

SF: 計算結果の16ビット目(MSB、最左端ビット)の値である。つまり、計算結果が負なら1、それ以外は0

ZF: 計算結果がゼロ (つまりすべてのビットが0) なら1、 それ以外は0

(注)教科書の表 2.2 で窺われるように、SF=0 かつ ZF=0 なら正数、ZF=1 ならゼロ、SF=1 であれば負数である。

o. 記憶装置には、処理中のプログラムやデータを記憶する () と、大量のデータを長期間保存しておく () がある。(教科書図 2.1)

(注) 補助記憶装置は、場合によっては入出力装置に分類される。

コンピュータ内の接続の形(何がどこに接続しているか)からは、補助記憶装置は入出力装置と見なす方がよい。

(注) 記憶装置では、情報は通常0/1で記憶されている。0/1の形になっていない情報、たとえば文字や画像、音声などは、0/1の形に変換して記憶される。

p. 入力装置・出力装置は、コンピュータが外部世界と情報をやり取りするための手段であり、人間が文字を入力する ()やコンピュータが文字を出力する()・()などがある。

[2] ノイマン型コンピュータ (教科書 2.1.1)

a. 現在広く使われているコンピュータの方式は、ノイマン型であるが、その特徴は (教科書 2.1.1 参照)

* プログラム内蔵方式 コンピュータの動作がプログラムによって制御され、そのプログラムは()に蓄積される

* 逐次処理方式 コンピュータは、プログラム中の命令を()に実行する

* 単一メモリ方式 プログラム(命令)もデータ(計算の対象となるデータ)も、()メモリ内に置かれている

b. 電卓はノイマン型とは言えないが、その理由は何か?

c. 人間の脳における情報処理は、多数の神経細胞によって同時並列に処理をされると考えられている。これはノイマン型と言えないが、その理由は何か?

d. プログラム内蔵方式が、内蔵しない方式に比較して、優れている点を2つ挙げよ

1)

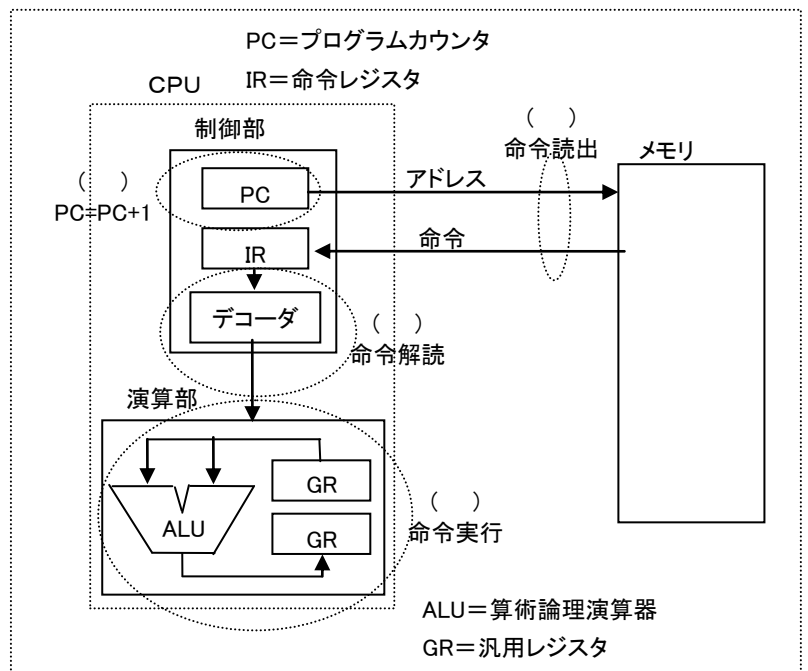
2)

[3] ノイマン型コンピュータの基本動作 (教科書 2.2)

- a. ノイマン型コンピュータでは、()に格納されている命令を1つ()て、()して、()する。これを繰り返して、逐次処理を進める。(教科書 2.2.1)
- b. 命令(機械命令、機械語命令とも呼ぶ)は、(正しいものを選び)
- ①ユーザがプログラムを起動する指示のことである(Windows ではアイコンのダブルクリックで代用する)
 - ②機械を細部にわたって動かす(制御する)ための最小単位の指示である
 - ③Java言語やC言語の最小単位の文のことである
 - ④Java言語やC言語のメソッドや関数(汎用の言葉で手続きと呼ぶ)のことである。
- c. 機械語命令は(正しいものを選び)
- ①ユーザが与えるので、ユーザの頭の中にある
 - ②補助記憶装置(ハードディスクやCDROMなど)に置いてあって、そこから取り出す
 - ③主記憶(メインメモリ)に置いてあって、そこから取り出す
 - ④CPUに置いてあって、そこから取り出す
- d. 1つ1つの機械語命令は(正しいものを選び)
- ①人間が読めればよいので自然言語で書かれている
 - ②Java言語やC言語など、人間も機械も読めるプログラミング言語で書かれている
 - ③機械が読めればよいので、0/1の列(バイナリ情報)で書かれている
 - ④機械が読めればよいので、電気信号(電圧の高低)で書かれている
- e. 機械語命令によるプログラムは(正しいものを選び)
- ①1つの機械語命令から成る
 - ②1つの機械語命令と一連のデータからなる
 - ③多数の機械語命令が並んだものから成る
 - ④多数の機械語命令が並んだものと一連のデータからなる
- f. 教科書の図 2.10 は「命令サイクル」を図示している。命令サイクルは、メモリ上に並んでいる命令に対して、上記 a. の「1つの命令の実行」を、次々と(逐次)行うものである。命令サイクルを正しく表したものを選び
- ① 命令解読 ⇒ 命令分析 ⇒ 命令実行 ⇒ PC を+1
 - ② 命令取出し ⇒ 命令解読 ⇒ 命令実行 ⇒ PC を+1
 - ③ 命令取出し ⇒ 命令実行 ⇒ 命令解読 ⇒ PC を+1
 - ④ 命令解読 ⇒ オペランド計算 ⇒ 命令実行 ⇒ PC を+1
- g. 前問で登場したサイクルの項目のうち、正しいものについて、右側の正しい説明を結び
- | | |
|------------|---------------------------------|
| 命令解読 | 命令を主記憶から取り出す |
| 命令分析 | 命令に指示された内容を実行する |
| 命令実行 | 命令を主記憶に書き戻す |
| 命令読出し(取出し) | 命令が何をすべきかを分析し、必要な信号を各所に出す |
| オペランド計算 | 命令がどのように使われるかを解析し、その情報を主記憶に書き戻す |
| | 命令中のオペランド指示を解釈しアドレスを計算する |
| | 命令に指示された内容を分析して実行する |
- h. 命令の側から見ると、メモリ上に置かれた命令は、実行サイクルによって次々と実行される。正しいものを選び
- ① 命令はメモリ上に1つだけ置かれ、それをCPUが実行する
 - ② 命令はメモリ上にたくさん置かれ、順番に関係なくあちこちにばらまかれている。CPUはそれを秩序正しく順番に実行する
 - ③ 命令はメモリ上にたくさん置かれているが、実行される順番に従っておかれている。CPUはそれを上から(アドレスの若い方から)順番に実行する
 - ④ 命令はCPUの中に置かれているが、それを直前の命令に指示された順番に従って次々と逐次実行する

i. 右図の中で、カッコを付した事がらが起きる順番に、①～④の番号を付けよ。

j. COMET II の加算命令 ADDA を実行するときにかかる事柄(事象)を、起こる順序に従って、具体的に書け。(教科書 2.2.2 の(2))



[4] 機械語命令とその形式 (教科書 3.1)

a. 機械語命令に関する次の説明で、正しいものには○、間違っているものには×をつけよ

- ① 機械語命令は CPU が何をするかを 0/1 の列で書いたもので、機械が読みとって実行する
- ② ニーモニックコードは、人間がプログラムを書くためのアセンブラ言語の一部で、いくつか(複数)のニーモニックコードを集めて1つの機械語命令に翻訳する
- ③ アセンブラ言語は、機械語命令レベルでプログラムを書くための方法で、原則として1行が1つの機械語命令に対応し、その機械語命令をニーモニックコードで記述する

- ④ Java などの(高級)プログラミング言語は、1つの文や行が1つの機械語命令に対応するということではなく、一般には1つの文は複数の機械語命令の組合せで実現される。

b. 機械語命令に関する次の説明で、正しいものには○、間違っているものには×をつけよ

- ① 機械語命令はC言語や Java 言語と同様に標準化が進んでおり、どの機種 of CPU においても同一である
- ② 機械語命令はCPUの機種によって異なるので、同じ機械語プログラムを共通に使うことはできない
- ③ C言語や Java 言語のプログラムは、機械語命令としてそのままCPUが実行することができる
- ④ C言語のプログラムは、一旦機械語命令に変換(翻訳)したものをCPUが実行する

(注) Java 言語のプログラムはやや特異で、一旦共通の仮想機械語に変換した後、それをそれぞれのCPUが(ハードウェアではなくて、ソフトウェアの)インタプリタによって実行する

c. 次の空欄を埋めよ

機械語命令は、メモリ上に置かれてCPUが解釈・実行する時は()の形で表されている。なぜなら、それが機械にとって最も都合がよい(メモリに記憶する時に記憶しやすい・CPUで実行する時解釈しやすい)からである。しかし、人間にとっては非常に扱いにくいので、機械語命令を()と呼ばれる記号に対応させて表す。

d. ニーモニックコードの説明で、正しいものには○を、間違っているものには×をつけよ

- ① 1つのニーモニックコードは、1つの機械命令の種類(命令コード、OP コード)を、数字(0/1 列)で表したものである
- ② 1つのニーモニックコードは、1つの機械命令の種類(命令コード、OP コード)を、文字列で表したものである
- ③ 1つのニーモニックコードは、1つの機械命令全体(命令コード+オペランド)を、文字列で表したものである。
- ④ ニーモニックコードは、Cや Java 言語などの高級言語で構文要素として使われている

e. 1つの機械命令は、大別して () と () の2つのフィールドから成っている。

それらのうち、() はその命令が「何をするか」、つまり操作の種類を区別するもので、() はその命令が操作する()を記述している。(教科書図 3.2)

f. 命令コード部分は、たとえば教科書の表 3.1 のようなものがある。表の左側の欄に書かれている表現(たとえば「ロード・LoaD」)はその命令コードのフルの名称、右側の欄に書かれている表現(たとえば「LD」)は、()である。

g. オペランドの説明で、正しいものを選び

- ① オペランドとは、演算の対象となる変数や定数を指定するものである

②オペランドとは、演算の種類(加算・減算など)を表すものである

③オペランドとは、演算に於て(どこから・どのようなデータを持ってきて・どのように計算して・どこへしまうか)を記述する

④オペランドとは、命令の長さを指定するものである

h. オペランドの具体的なイメージは、たとえば算術加算命令(教科書の表 3.1 の4番目に書かれている)であれば、何と何を足すのか(足し算は2つの入力を指定する必要がある)、更にはその結果をどこへしまうか(出力は1つでよい)、を指定する。足し算では、これらの「入力」と「出力」(合計3つ)を「オペランド」として指定する必要がある。これらの入力と出力は、データを置いておけるところ、つまり、CPU内の()か、記憶装置内の()になる。

[5] オペランドの指定方法 (教科書 3.1.2)

a. 教科書の 3.1.2 にははっきりとは書いていないが、オペランド(操作の対象)の指定をするには、

① 汎用レジスタの場合は、汎用レジスタの番号 (COMET II の場合はGR0~GR7までのGRの番号)

② 主記憶(メインメモリ)の場合は、主記憶のアドレス (COMET II の場合は0~65535番地)

のいずれかで指定することになる。たとえば、「AとBとを足してCに格納する」の場合、AとCをGR3とし、Bを主記憶の3524番地とする、といった指定をすることになる。

b. 教科書の 3.1.2 にあるように、オペランドの個数(オペランドに書くアドレスの個数)によって、いくつかの種類がある。

* 3オペランド命令は、2つの () アドレスと、1つの () アドレスを記述する方式で、たとえば加算命令で $C=A+B$ の場合、AとBとCのアドレスを記述する。

* 2オペランド命令は、 $C=A+B$ の3つのうち、入力の1つ(たとえばA)と出力Cを同一のものに限定する方式である。つまり、必ず $A=A+B$ の形式でなければならない、とする。

* 1オペランド方式は、2オペランド方式 $A=A+B$ を前提とし、更に、2つのうちAを、(汎用レジスタではなく)特定のレジスタ(アキュムレータ)に固定する方式である。つまり、A側は固定されるので指定する必要がなく、命令上はBのみを指定すればよい。

* 0オペランド方式は、教科書図 3.6 にあるように、演算をスタックを用いて行う方式で、この場合は一切オペランドの場所を指定する必要がなくなる。

c. 教科書 3.1.2 の最後に、「即値」がある。即値はオペランドに操作データを()する。

d. 教科書 3.1.2 の最後に、「すべての命令の長さ(ビット数)が一定である()と、命令によって長さがkとなる()がある。」としている。

《解答》

[1] a. 演算(装置) 記憶(装置) 入力(装置)と出力(装置)または併せて入出力(装置) 制御(装置)

b. 中央処理装置 CPU (Central Processing Unit の略)

c. 中央処理装置 記憶装置 入出力装置

d. 演算装置 制御装置 (bの逆を問うているだけ)

e. 右図

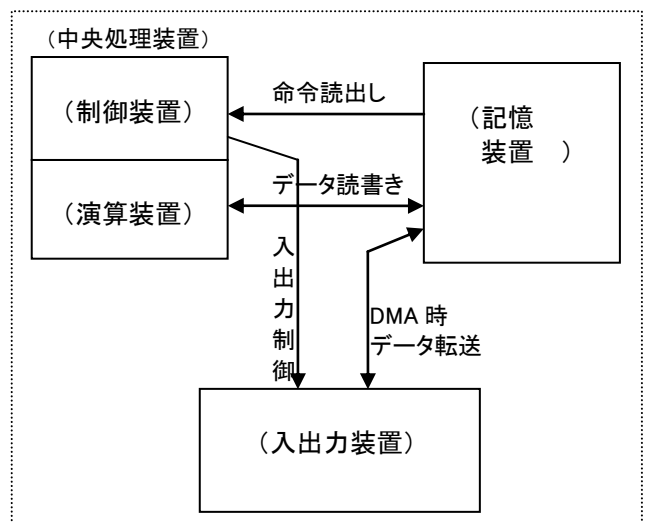
f. ①と④

②は「メモリを含む」が誤り

③は、メモリは、データとプログラム(命令)が置かれる

g. ②

h. ② 順番を決めるのも、解釈実行するのも制御装置



- i. ③と④
- j. 汎用レジスタ 算術論理演算装置 フラグレジスタ
- k. Arithmetic and Logical Unit 算術論理演算装置
- l. General Register

m. CPUによって異なる

- n. OF SF ZF

OF: 演算結果がある範囲を超えると1(オーバーフロー)、超えなければ0

SF: 演算結果の符号が負のとき1、それ以外(正か0)のとき0

ZF: 演算結果がゼロの時1、それ以外の時0

- o. 主記憶装置(メインメモリ) 補助記憶装置

- p. キーボード ディスプレイ プリンタ

—(データを保存する、命令を保存する)、 制御装置—命令を解釈してどのような処理を行うか決める

演算装置—計算(四則演算など)を行う、 汎用レジスタ—計算結果を一時的に保存する

[2]

- a. メモリ(メインメモリ・主記憶) (注: コンピュータで実行される命令は必ずメインメモリ上になければならない)

順次(順番に) (注: メインメモリには番地(アドレス)が先頭から 0, 1, 2, ...のように付いており、その順に実行する)

同じ (注: 命令格納用メモリとデータ格納用メモリとに、分けることをしないのが原則。但し最近では2つの理由から分割することがある。1つは性能上の理由〜教4章ハードウェアアーキテクチャ、もう1つはプログラムを誤って書き換えてしまうことがないようにするというソフトウェア管理上の理由である)

- b. 一般用の電卓ではプログラム内蔵ではない。逐次処理でも単一メモリでもない(プログラム自体がないのだから)。

- c. 逐次処理ではない(同時並行に複数の処理をする)。

- d. 点1) プログラムを外に置く極端な例は、電卓を人間が使う場合で、人間が「次の操作」を手で指定する。この場合、プログラムを電卓に与える速度(人間がキーを押す速度)が遅いため、計算が遅くなる。これに反して、プログラム内蔵だとプログラム(個々の命令)を高速に読み出せるので、処理が高速化される。

点2) プログラムをメモリに内蔵しない例として、配線によってプログラムを与える方式がある。教科書 1.2.3にある ENIAC は配線プログラム方式であったし、20年ぐらい前のエレベータの制御はやはり配線プログラムであった。この方式の欠点は、プログラムの変更・入替えが困難または時間がかかり、単一用途ではよいのだが、今日見られる多用途・万能のコンピュータとはならない。

- [3] a. 主記憶(メインメモリ、メモリ) 取り出して 解釈して 実行する

- b. ② c. ③ d. ③

- e. ④ (注: プログラムは、命令の列と、その命令で操作されるデータから成る) ③だと不足

- f. ② (注: 取出しは「読出し」とも言う。 解釈は「解釈」、「デコード」とも言う。)

- g. 命令解釈 —— 命令が何をすべきかを分析し、必要な信号を各所に出す

命令分析 サイクルにはない

命令実行 —— 命令に指示された内容を実行する

命令読出し(取出し) —— 命令を主記憶から取り出す

オペランド計算 サイクルにはない (注: 実行の一部として行われる)

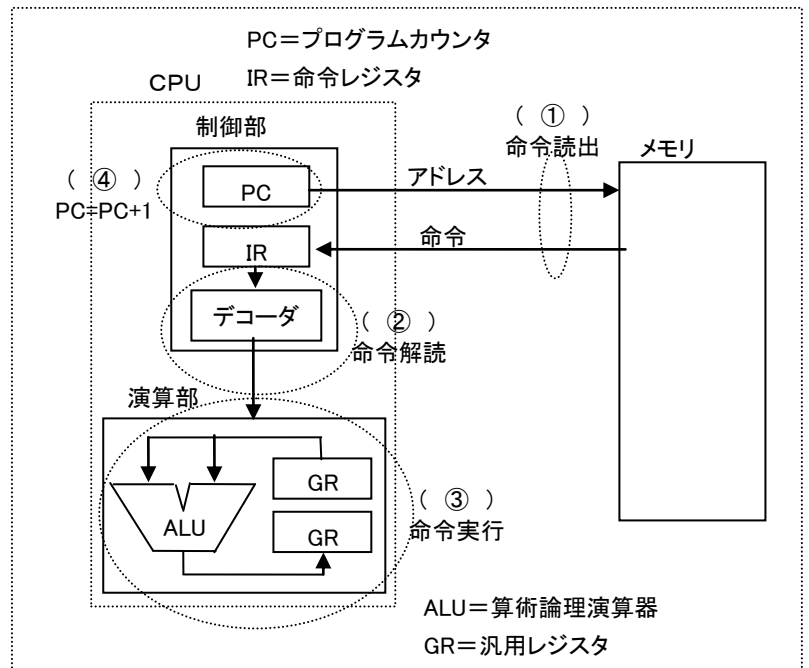
- h. ③ 正確には、格段の指示がない限りアドレスの順番に実行する。ジャンプ命令(分岐命令)があると、実行は指示された場所へ移る

- i. (左図)

- j. (略)

[4]

- a. ①○ ②× ③○ ④○
b. ①× (CPU 機種ごとに異なる)
②○ (CPU 機種ごとに別々のプログラムになる)
③× (④で言うように、翻訳する必要がある)
④○
c. 0/1 ニーモニックコード
d. ①× ②○ (命令コード=OPコード、同じもの)
③× (オペランドは含まない)
④×
e. 命令コード(別名OPコード) オペランド
命令コード オペランド 対象データ
f. ニーモニックコード
g. ①と③が正しい 演算の種類を表すのは命令コードである。命令の長さを指定する情報は書かれないのが普通。
h. 汎用レジスタ 主記憶(メインメモリ)



[5]

- b. 入力(となるオペランドの) 出力(となるオペランドの)
c. 直接記述する (注: 即値命令では、たとえば、汎用レジスタに指定した値をセットする、といった使い方ができる)
d. 固定長命令方式 可変長命令方式
(注: 可変長命令と言うのは、命令の長さが実行中に変わるという意味ではない。それぞれの命令の長さは機械として決まっているのだが、命令の種類によっていろいろな長さの命令がある、という意味である。たとえば加算命令はオペランドの数が3つ必要なので命令が長いがリターン命令はオペランドが不要なので短い、などである。)

《基本情報処理技術者試験問題から関連問題》

1) コンピュータは、入力、記憶、演算、制御及び出力の五つの機能を実現する各装置から構成される。命令はどの装置から取り出され、どの装置で解釈されるか。(基本 18 春 26)

	取出し	解釈
ア	演算	制御
イ	記憶	制御
ウ	制御	演算
エ	入力	演算

2) コンピュータの基本アーキテクチャで、プログラムとデータを一緒にコンピュータの記憶装置の中に読み込んで実行する方式はどれか。(基本 15 秋 29)(基本 17 秋 26)

- ア アドレス方式 イ 仮想記憶方式 ウ 直接プログラム制御方式 エ プログラム内蔵方式

3) プログラムを主記憶に読み込んでおき、CPU が順次読み出し実行する方式はどれか。(基本 19 秋 26)

- ア アドレス方式 イ 仮想記憶方式 ウ 直接プログラム制御方式 エ プログラム記憶方式

4) プロセッサは演算機構及び制御機構からなる。制御機構に分類されるものはどれか。(基本 16 春 29)

- ア アキュムレータ イ 加算器 ウ 補数器 エ 命令デコーダ

5) プロセッサの制御機構に分類されるものはどれか。(基本 19 春 26)

- ア ALU イ アキュムレータ ウ 命令デコーダ エ メモリアドレスレジスタ

6) コンピュータの命令実行順序として、適切なものはどれか。(基本 18 春 18)

- ア オペランド読出し → 命令の解釈 → 命令フェッチ → 命令の実行
 イ オペランド読出し → 命令フェッチ → 命令の解釈 → 命令の実行
 ウ 命令の解釈 → 命令フェッチ → オペランド読出し → 命令の実行
 エ 命令フェッチ → 命令の解釈 → オペランド読出し → 命令の実行

7) 主記憶へのアクセスを伴う演算命令を実行するとき、命令解釈とオペランド読出しの間に行われる動作はどれか。(基本 20 春 18)

- ア 実効アドレス計算 イ 入出力装置起動 ウ 分岐アドレス計算 エ 割込み発生

8) シングルチップマイコンの特徴として、最も適切なものはどれか。(基本 21 春 10)

- ア PC のメイン CPU に適している。
 イ ROM は内蔵されているが、RAM は内蔵されていない。
 ウ 高速処理システム又は大規模なシステムに適している。
 エ 入出力機能が内蔵されている。

《解答》

1) イ 2) エ 3) エ 4) エ 5) ウ 6) エ 7) ア
 8) (注) この問題は教科書 15 ページに一言触れている。「シングルチップ型は、組込み用マイコンに適するように CPU、メモリ、インタフェース、各種周辺装置(A-D 変換器など)の機能を一つのチップ内に構成したものである。一方、マルチチップ型は各機能を個別のチップで構成したものである。たとえば H8 はシングルチップ型、Pentium はマルチチップ型の CPU である」。

- ア: × PC のメイン CPU は、マルチチップ構成が使われるのが普通(上記 Pentium の例)
 イ: × シングルチップ型には、ROM だけでなく、RAM も含まれることが多い。RAM は(作業用)変数を置くのに必要になる。
 ウ: × 高速・大規模なシステムには、チップの面積容量の制約や拡張性の問題があるので、使われない。家電や車載などの組み込みの用途に使われることが多い。
 エ: ○ 教科書の「各種周辺装置」が入出力機能に当たる。家電・車載などコストダウンが求められるので、なるべく1つのチップに組み込むことが低廉化につながる